Amendments to the Claims

1   Claim 1 (currently amended): A method for programmatically enforcing referential integrity

2   constraints among associations between class instances, comprising steps of:

3   programmatically determining, upon a request to modify an association end for an

4   association between an instance of a first class and an instance of a second class, the association

5   comprising one association end from the instance of the first class to the instance of the second

6   class and another association end from the instance of the second class to the instance of the first

7   class, ~~when evaluating a request to set an association end to reflect an association from an~~

8   ~~instance of a first class to an instance of a second class,~~ whether the association end to be

9   modified has a single multiplicity or a many multiplicity;

10   if the association end to be modified has the single multiplicity, programmatically

11   performing the steps of:

12   disconnecting a previously-existing inverse association end of the association end

13   to be modified, if any;

14   setting a new inverse association end of the association end to be modified; and

15   modifying the association end for the association end to be modified,

16   wherein the steps of disconnecting, setting, and modifying are performed atomically; and

17   if the association end to be modified has the many multiplicity, programmatically

18   performing the steps of:

19   modifying the association end to be modified;

20   disconnecting a previously-existing association end of the association end to be

21   modified, if any; and

Serial No. 09/827,290   -2-   Docket RSW920000173US1

22   setting the inverse association end for the association end to be modified,

23   wherein the steps of modifying, disconnecting, and setting are performed atomically,

24   ~~setting the requested association end; and~~

25   ~~programmatically modifying an inverse association end of the association to reflect an~~

26 ~~inverse association from the instance of the second class to the instance of the first class, after~~

27 ~~disconnecting the inverse association end from an existing instance, if any;~~

28   ~~wherein an ordering of the setting step and the programmatically modifying step depends~~

29 ~~on an outcome of the determining step.~~

Claims 2 - 4 (canceled)

1   Claim 5 (currently amended): The method according to Claim 1, further comprising [[steps]] the

2   step of[[:]] serializing the association between the instance of the first class and the instance of

3   the second class by performing steps of:

4   determining whether the association end to be modified or the inverse association end is a

5   primary end of the association; and

6   serializing only the primary end of the association during [[a]] the serialization operation.

1   Claim 6 (currently amended): The method according to Claim 1, wherein the method is provided

2   as a single link helper ~~objects~~ object and a multiple link helper object for each association,

3   wherein the single link helper object performs the steps of disconnecting, setting, and modifying

4   for the single multiplicity association end and the multiple link helper object performs the steps

Serial No. 09/827,290   -3-   Docket RSW920000173US1

5   of modifying, disconnecting, and setting for the many multiplicity association end.

1   Claim 7 (currently amended): A computer program product for programmatically enforcing

2   referential integrity constraints among associations between class instances, wherein the

3   computer program product is embodied on one or more computer readable media and comprises

4   computer-readable program code means for:

5   computer-readable program code means for programmatically determining, upon a

6   request to modify an association end for an association between an instance of a first class and an

7   instance of a second class, the association comprising one association end from the instance of

8   the first class to the instance of the second class and another association end from the instance of

9   the second class to the instance of the first class, when evaluating a request to set an association

10   end to reflect an association from an instance of a first class to an instance of a second class;

11   whether the association end to be modified has a single multiplicity or a many multiplicity;

12   computer-readable program code means for setting the requested association end; and

13   if the association end to be modified has the single multiplicity, programmatically

14   performing the steps of:

15   disconnecting a previously-existing inverse association end of the association end

16   to be modified, if any;

17   setting a new inverse association end of the association end to be modified; and

18   modifying the association end for the association end to be modified,

19   wherein the steps of disconnecting, setting, and modifying are performed atomically; and

20   if the association end to be modified has the many multiplicity, programmatically

Serial No. 09/827,290   -4-   Docket RSW920000173US1

21   performing the steps of:

22   modifying the association end to be modified;

23   disconnecting a previously-existing association end of the association end to be

24   modified, if any; and

25   setting the inverse association end for the association end to be modified,

26   wherein the steps of modifying, disconnecting, and setting are performed atomically.

27   ———computer-readable program code means for programmatically modifying an inverse

28   association end of the association to reflect an inverse association from the instance of the second

29   class to the instance of the first class, after disconnecting the inverse association end from an

30   existing instance, if any;

31   ———wherein an ordering of operating the computer-readable program code means for setting

32   and the computer-readable program code means for programmatically modifying depends on an

33   outcome of the computer-readable program code means for determining.

        Claims 8 - 9 (canceled)

1   Claim 10 (currently amended): The computer program product according to Claim 7, further

2   comprising computer-readable program code means for serializing the association between the

3   instance of the first class and the instance of the second class by performing steps of:

4   computer-readable program code means for determining whether the association end to

5   be modified or the inverse association end is a primary end of the association; and

6   computer-readable program code means for serializing only the primary end of the

        Serial No. 09/827,290   -5-   Docket RSW920000173US1

7   association during [[a]] the serialization-operation.

1   Claim 11 (currently amended): A system for programmatically enforcing referential integrity

2   constraints among associations between class instances, comprising means for:

3   programmatically means-for-determining, upon a request to modify an association end for

4   an association between an instance of a first class and an instance of a second class, the

5   association comprising one association end from the instance of the first class to the instance of

6   the second class and another association end from the instance of the second class to the instance

7   of the first class, when-evaluating a request to set an association end to reflect an association

8   from an instance of a first class to an instance of a second class, whether the association end to be

9   modified has a single multiplicity or a many multiplicity;

10   if the association end to be modified has the single multiplicity, programmatically

11   performing the steps of:

12   disconnecting a previously-existing inverse association end of the association end

13   to be modified, if any;

14   setting a new inverse association end of the association end to be modified; and

15   modifying the association end for the association end to be modified,

16   wherein the steps of disconnecting, setting, and modifying are performed atomically; and

17   if the association end to be modified has the many multiplicity, programmatically

18   performing the steps of:

19   modifying the association end to be modified;

20   disconnecting a previously-existing association end of the association end to be

Serial No. 09/827,290   -6-   Docket RSW920000173US1

21   modified, if any; and

22   setting the inverse association end for the association end to be modified,

23   wherein the steps of modifying, disconnecting, and setting are performed atomically.

24   ~~means for setting the requested association end; and~~

25 ~~means for programmatically modifying an inverse association end of the association to~~

26 ~~reflect an inverse association from the instance of the second class to the instance of the first~~

27 ~~class, after disconnecting the inverse association end from an existing instance, if any;~~

28 ~~wherein an ordering of operating the means for setting and the means for~~

29 ~~programmatically modifying depends on an outcome of the means for determining.~~

    Claims 12 - 13 (canceled)

1   Claim 14 (currently amended): The system according to Claim 11, further comprising means for

2   serializing the association between the instance of the first class and the instance of the second

3   class by performing steps of:

4   ~~means for~~ determining whether the association end to be modified or the inverse

5   association end is a primary end of the association; and

6   means for serializing only the primary end of the association during [[a]] the serialization

7   ~~operation.~~

1   Claim 15 (new): The method according to Claim 1, wherein one or more structured markup

2   language representations specify instances of the first class, instances of the second class, and

    Serial No. 09/827,290   -7-   Docket RSW920000173US1

3   associations between the instances of the first and second classes.

1   Claim 16 (new): The method according to Claim 15, wherein only one association end for each

2   association between instances is specified in the structured markup language representations.

1   Claim 17 (new): The method according to Claim 16, wherein the only one association end is an

2   association end designated as a primary end for the association.

1   Claim 18 (new): The method according to Claim 15, wherein a serialization of results of the

2   request to modify the association end that has the single multiplicity comprises the step of:

3   determining whether the association end to be modified is a primary end for the

4   association, and if so, programmatically performing the steps of:

5   removing the representation of the previously-existing inverse association end, if

6   any, from the structured markup representation in which it is specified; and

7   adding a representation of the new inverse association end.

Serial No. 09/827,290   -8-   Docket RSW920000173US1